DEVELOPING AN AGRICULTURAL STOCK CONTROL SYSTEM 'FARMWARE'

Zheng Bitao, Faizan Qamar

Faculty of Information Science & Technology
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor

Abstrak

Pembangunan Aplikasi Mudah Alih Farmware memberi tumpuan kepada membina satu sistem yang kemas dan lancar bagi membolehkan pengguna mengurus data inventori dengan mudah serta meningkatkan produktiviti pertanian. Aplikasi ini direka khusus untuk membantu individu bebas dan petani berskala kecil yang tidak bernaung di bawah organisasi pertanian besar tetapi masih mempunyai hasrat atau keperluan untuk menjalankan aktiviti pengeluaran pertanian. Salah satu ciri utama aplikasi Farmware ialah mod luar talian, yang membolehkan pengguna merekod, menyunting dan menyemak laporan inventori serta produktiviti walaupun tanpa sambungan internet. Ciri ini menjadikan aplikasi ini amat praktikal untuk kawasan luar bandar atau kawasan pertanian terpencil yang mempunyai kekangan capaian internet. Dengan mendigitalkan pengurusan inventori dan memudahkan pencatatan data, aplikasi Farmware bertujuan untuk membantu perniagaan kecil serta menyokong proses membuat keputusan dan pengurusan tanaman. Proses pembangunan melibatkan peringkat seperti pengumpulan keperluan, reka bentuk sistem, pelaksanaan, dan ujian bagi memastikan aplikasi ini berfungsi dengan baik dan mesra pengguna.

Kata kunci: Pertanian, Aplikasi Mudah Alih, Pengurusan Inventori, Mod Luar Talian, Produktiviti

Abstract

The Farmware Mobile App development focuses on build a clean and fluent system to provide users with convenient to manage the inventory data, and to boost the agricultural productivity. This application specifically designed to help independent individuals and small scale farmers who may not be affiliated with large agricultural organizations but still have the intention or need to engage in agricultural production. One of the key features of the Farmware app is its offline mode, which allows users to record, edit, and browse inventory and productivity reports even without an internet connection. This makes the app especially practical for rural or remote farming areas where connectivity is limited. By digitalizing inventory management and making data recording more convenient, the Farmware app seeks to help small business and to support decision making and crop management. The development process includes stages such as requirements gathering, system design, implementation, and testing to ensure the application is both functional and user-friendly.

Keywords: Agriculture, Mobile App, Inventory Management, Offline Mode, Productivity

1.0 INTRODUCTION

Agriculture is considered one of the most important factors that ensuring food security and supporting economic development. However, small-scale agricultural producers still face many challenges in managing farm data, accessing technology, and maintaining competitiveness in the market. Most digital systems available today are expensive or too complex for these users. To fix this issue, this project aims to develop a stock control system specifically to support the agricultural industry. By supporting small businesses, this can increase the power of the community.

Currently, most advanced systems are expensive or not suitable for small business. Therefore, creating a flexible, customizable, and lightweight solution can be highly beneficial for these businesses. This project intends to deliver a functional information management system that has the potential to expand across multiple

platforms. With Android support, users can access and input critical data related to crop status, fertilizer stock, and other essential production details using their mobile devices.

With the increasing world's population, agri-food demands are growing — posing the need to switch from traditional agricultural methods to smart agriculture practices (Abbasi et al., 2022). In this progress, there shall be many new problems and requirements on developing raises, this project is aimed to solve these needs. Furthermore, as the resources of the agriculture industry today are still mainly distributed among large businesses such as Corteva Agriscience and Kuala Lumpur Kepong Berhad, there should be more attention upon small business, which could contribute a lot of productions and motivates the community and personal funders. The importance of decentralization is just as Josephine Phillip Msangi described: “Eradicating extreme hunger and poverty depends on improving agriculture and enacting policies that support small-scale agricultural producers’ productivity and strengthening food processing and fortification.”(Msangi & Msangi, 2014).

Information technology is still in demand for the agricultural industry. Even though many products and services already exist, more facilities are still needed. L J E Dewi1, I N S W Wijaya1 and K A Seputra1 have described that, "The surplus of crops in the harvest time makes the farmers lose since the sale price of their product is very low. This is also the case with unclear distribution paths that cause a loss to the local farmers since Buleleng local products may be bought before the harvest time at a low price, then harvested and brought outside of the region, and are sold again in Buleleng with labels of products that indicate that the products are from outside of Buleleng regency and the products are sold in Buleleng at a high price"(Dewi et al., 2021).

To respond to these ongoing issues, this project provides a platform that empowers small producers with accessible technology and practical functions for data and stock management. With further development, the system may expand into broader features like real-time environmental data, by integrating with some sensors and embedded system, and may support barcode system. Farmware strengthens the role of small-scale agriculture in national and global food productions.

2.0 LITERATURE REVIEW

Agricultural production needs to manage various kinds of important data such as seed use, fertilizers, and trade records. Without good and timely information about stocks, crop yields can be affected. Because of that, a reliable inventory management system is needed. According to Madamidola (Madamidola et al., 2024), inventory management is very important in any business that handles physical goods, like in manufacturing or retail. It helps avoid too much or too little stock, which saves costs and keeps operations running. Agriculture has similarities with those industries, so a system that is simple and stable would be useful.

In earlier years, not many smart systems were discussed. Hasan (Hasan et al., 2013) talked about Management Information Systems (MIS) that collect internal data and summarize it into reports to help in decisions. But these systems were not like today's systems that use smart features or automation. Now, data mining and artificial intelligence are more common. Oladele (Oladele et al., 2021) explained that data mining, also called KDD (Knowledge Discovery in Databases), involves things like machine learning and statistics to find patterns in large data. This shows that technology is moving toward smarter systems.

Security also becomes a concern when systems get more complex. Hameed & Arachchilage (Hameed & Arachchilage, 2021) said that self-efficacy is a factor that often shows up in studies about adopting security technology in information systems, but its effect is still not fully clear. This might be something to study more in the future.

In terms of inventory models, the Economic Order Quantity (EOQ) model has been used in many sectors, like retail and pharmaceuticals. Agarwal (Agarwal, 2014) said that EOQ helps figure out the best time and amount to order, especially when the business is stable. Another model is Just-In-Time (JIT), which tries to reduce how much stock is stored and instead orders things based on real demand. For small agricultural businesses that don't have a stable customer base, JIT might be more useful. So this project will try to follow that idea.

Some existing tools already offer inventory solutions, but they aren't always suitable. For example, Granular is known as one of the top Farm Management Information Systems (*6 Farm Information Management Systems Changing The Future of Farming*, 2022), and it avoids the use of spreadsheets. However, it requires users to register with a Business Partner ID and has subscription costs that can reach up to $5000 per year. This may stop small farmers or people who just want to try farming for charity or personal reasons, especially in poorer areas.

FarmLogs and Trimble Ag Software also have their own advantages and disadvantages. FarmLogs works well for entering crop data and weather info, but it depends too much on internet access. Trimble has good GPS and monitoring features, but it's expensive and can be hard for beginners to use. Table 1 compares these tools and shows their problems and possible fixes.

Table 1 Existed applications comparing

Name	Advantage	Disadvantage	Problem	Solution
Granular	Granular provides a clean, spreadsheet-free interface, making it easy to navigate and visually appealing.	The annual subscription fee can be prohibitive for small farmers or charitable agricultural efforts.	The high cost and complex registration process for farmers in low-income areas or individuals interested in agricultural affairs or charity work.	Offering a basic free version or a lower-cost tier could make the system more accessible. Reducing the number of mandatory registration fields or adding options for personal users.
FarmLogs	FarmLogs provide efficient data input on crops, field activities, and weather.	Users in rural areas with limited connectivity may struggle to access real-time and data syncing.	While a free version exists, it may not provide all the tools needed for comprehensive farm management, which can make difficult to use.	Adding offline data entry and syncing options for rural areas would improve usability for users with limited internet access.
Trimble Ag Software	Trimble Ag Software is strong in	The advanced precision agriculture tools	The precision tools require knowledge and experience,	Providing a more basic user interface with

precision agriculture features, offering GPS guidance, and monitoring.	are most valuable to larger operations with complex needs, which may overwhelm smaller or novice farmers. It also not cheap.	which can be challenging for small business farmers.	simplified options could attract smaller farmers. Offering training resources or tutorials could help users make better use of the advanced features.

Some recent studies show that machine learning and cloud computing are becoming more used. Amini & Rahmani (Amini & Rahmani, 2023) introduced the WEKA tool, which allows testing different machine learning techniques on data. They also showed how collaboration between agriculture experts and machine learning researchers is important. Akhter & Sofi (Akhter & Sofi, 2022) said that machine learning and data analysis can help raise crop yields and reduce the workload for farmers.

This project will try to include a bit of automation. For example, the system could check user data and give out some simple tags or short notes, so users can quickly know what's happening, without needing to read everything in detail. If the user wants more info, they can ask the system to show a full report based on what they choose. This should make things easier to use, especially for people who are new or running small farms or businesses.

To sum up, this review talked about why inventory systems in agriculture are important and showed some of the tech that already exists. It mentioned models like EOQ and JIT, and also some tools like Granular. But these tools still have problems for new users or small groups. So this project will try to offer something simpler, lighter, and easier to start with, especially for people who are just trying to start farming or dealing with difficult situations.

3.0 METHODOLOGY

This section outlines the methods used in the development of Farmware, an agricultural inventory management system targeted at small-scale farmers, individuals, and amateur

agricultural producers. The project follows a structured development approach aligned with both user needs and national goals set by the Ministry of Agriculture and Food Security of Malaysia (*Profile*, 2024).

3.1 Needs and Objectives

This project is made to help small farmers or people who want to do farming but don't have good tools. Many of them have problems like not having fast internet, not being able to pay for expensive apps, or not knowing how to use complicated systems. So, this system tries to be simple, cheap, and still can work without internet. This also follows the goal of the Ministry, which wants to improve farming in a planned and complete way.

Besides just solving the direct problems, this project also looks at general things like what's happening in the IT world, what new tech is used, and what users really deal with when using apps. These ideas are used to make sure the system is useful, especially for people who are new to farming or don't have much support.

3.2 Design and Development Approach

The project uses normal software development steps. First comes planning, then design, then building, testing, and improving. While making the system, the plan and design are always checked and changed if needed based on feedback and understanding.

The design is not only based on technical functions but also based on how users really work and what they need. Some articles were used to help guide this. For example, Ebbers (Ebbers et al., 2022) said that good documentation makes systems easier to reuse and understand later. So, writing and recording the design clearly is also part of this work. The design covers things like how the system is built, how the screens look, and how the data is handled.

3.3 Key Features and Requirements

The main feature of this app is to help manage farming items like seeds, crops, fertilizers, and tools. The system works offline using local database (SQLite), so users can input and change data without internet. When the connection comes back, the data can be synced to the server.

Later, if possible, the system will try to give advice based on what the user stores, like telling them what to restock or what product might be good to sell. If the user is online, the app may also show extra info like weather or market prices. Tata Sutabri (Tata Sutabri, 2023) mentioned that the internet helps people get information no matter where they are, so adding that to the app could be helpful too.

3.4 Tools and Technologies

The system is made using Android Studio for the mobile app and SQLite for offline data. Tools like Visual Studio Code and GitHub are also used to help write code and manage files. The app is made for Android phones, which are more common and easier to get.

Some extra tools like JUnit or cloud platforms might be used if the project grows bigger. The system is planned to be usable by itself (offline), but also ready to connect with online services if needed. Cloud services like AWS might be used for storing or syncing data in future.

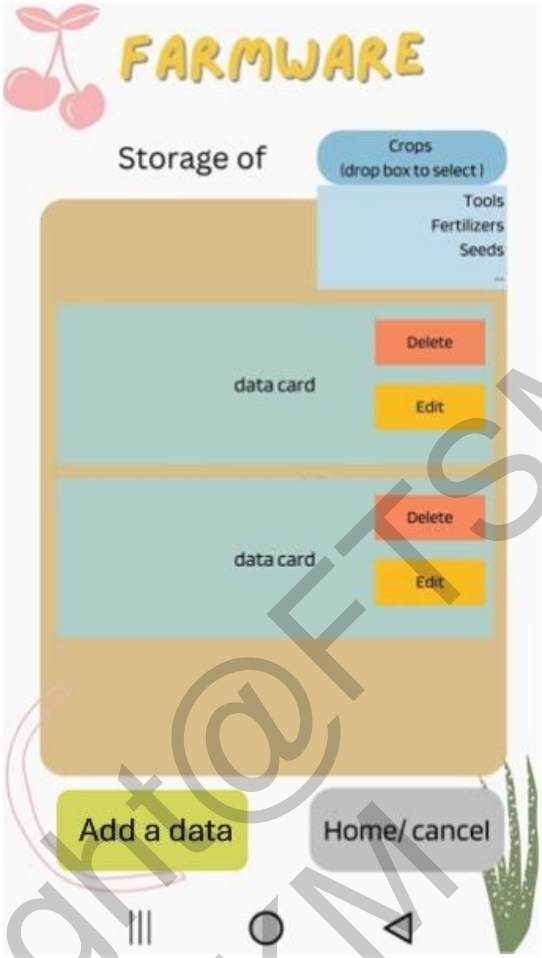


Figure 1 UI design browsing data

4.0 RESULTS

4.1 Application Development

The Farmware app was developed step by step. First, the basic system structure was created. This includes important files like MainActivity, ChooseMode, and Utils. MainActivity does not appear on the screen but handles setup tasks such as inserting test data and switching to offline mode.

After that, the user interface and login system were built. The login system uses fragments to show Login, Signup, and Forgot Password screens. The password reset system uses base64 encoding. This allows the admin to generate a new password manually and send it to the user.

After login, the user enters the Home screen. From there, the user can open the data form, view reports, give feedback, or edit their profile. The form section includes Add and Edit pages that check user input and save it into the database. These are two of the more complex parts of the system.

The app uses SQLite with Room to save data on the phone. This lets users use the app even when there is no internet.

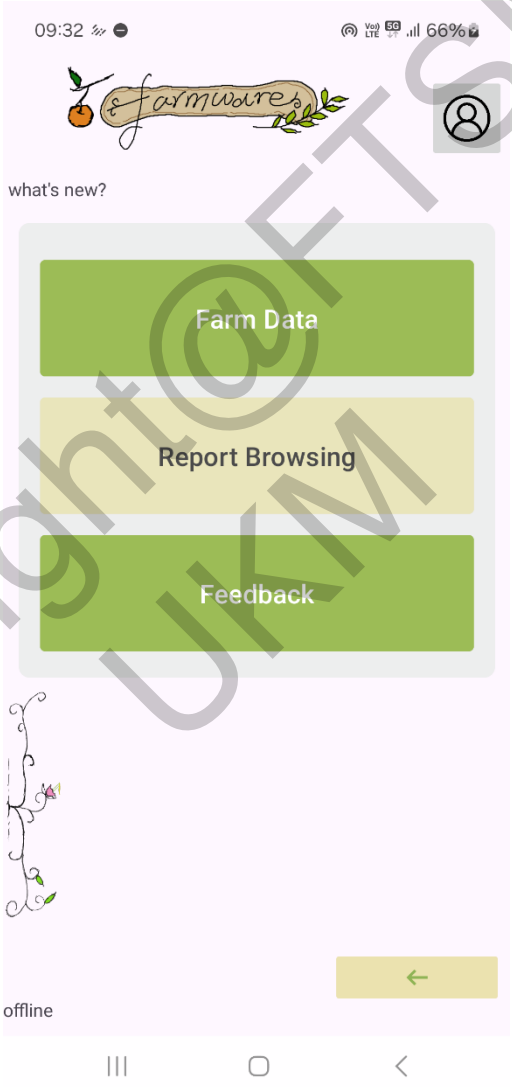


Figure 2 Home screen interface

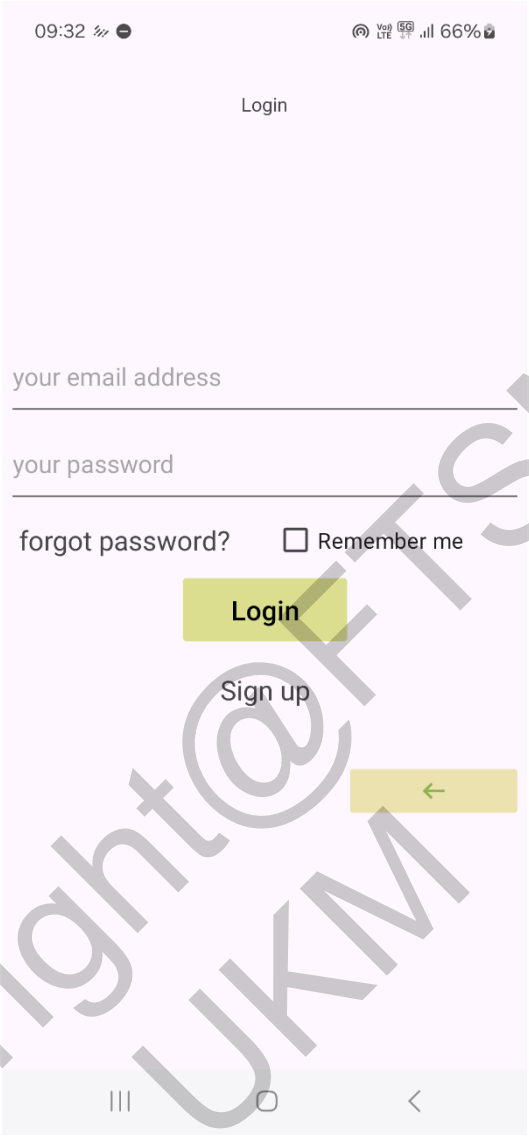


Figure 3 Login screen

Figure 4 Add Data form screen

Figure 5 Edit Data form screen

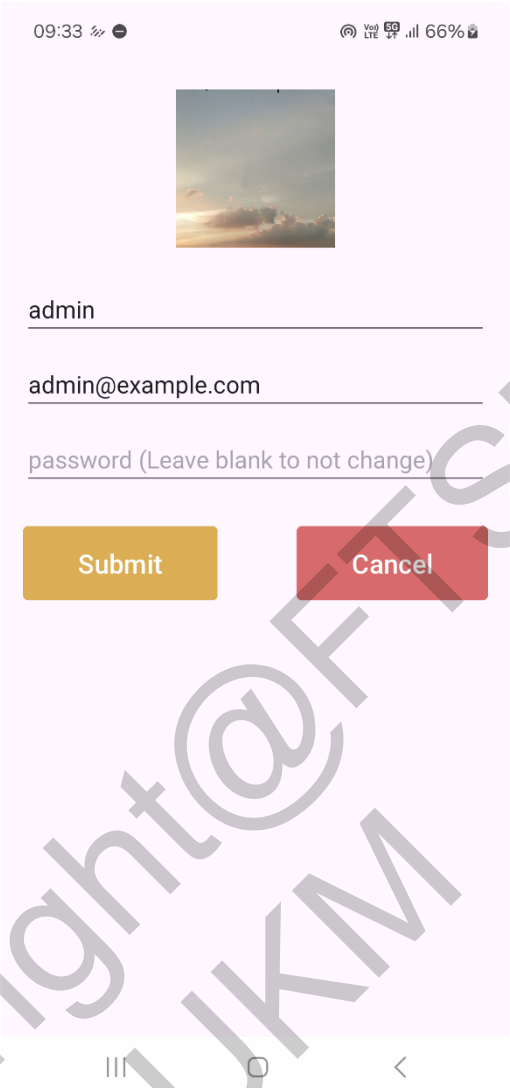


Figure 6 User Profile page

```
// in add new data
import android.content.Context
import android.content.Intent
import java.text.SimpleDateFormat
import java.util.Date
import java.util.Locale
import android.net.Uri
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity

// in database
import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase
import com.example.farmware.offlinebase.daos.*
import com.example.farmware.offlinebase.entities.*
```

Figure 7 Important packages involved

```python
#!/usr/bin/env python3
import base64

# Must match the XOR key used in the app's encryptPassword()
KEY = ord('K')

def decrypt(enc_str: str) -> str:
    """Decrypts a Base64-encoded, XOR-obfuscated string and restores original order."""
    raw       = base64.b64decode(enc_str)
    # XOR each byte, build the decrypted string
    decrypted = ''.join(chr(b ^ KEY) for b in raw)
    # reverse it to correct the order
    return decrypted[::-1]

if __name__ == "__main__":
    try:
        enc = input("Enter encrypted password (paste here): ").strip()
        if not enc:
            print("No input provided. Exiting.")
        else:
            dec = decrypt(enc)
            print("\nDecrypted password:")
            print(dec)  # copy-friendly output
    except Exception as e:
        print(f"Error during decryption: {e}")
    finally:
        input("\nPress Enter to exit.")

```

Figure 8 Admin password recover script using base64

Some features still need improvement. Right now, the CRUD logic is written separately for each table. This means that the code is longer and harder to maintain in the future.

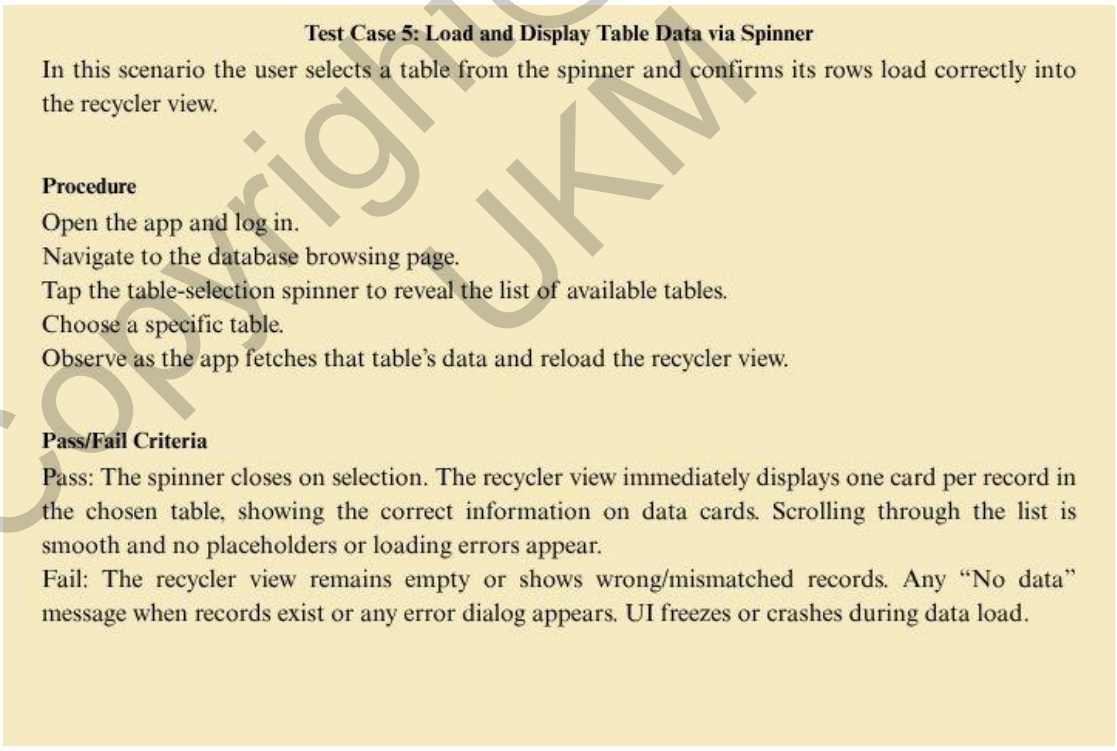
4.2 Application Evaluation

i. Test Plan

The test plan included unit tests, feature tests, and full system tests. These tests checked if features like login, form validation, offline data saving, and password reset worked properly. Testing also covered app behavior during install, upgrade, and logout.

ii. Test Case Design

Test cases were created to check each important feature. They include both user actions and expected system responses.



Test Case 5: Load and Display Table Data via Spinner

In this scenario the user selects a table from the spinner and confirms its rows load correctly into the recycler view.

Procedure

Open the app and log in.
Navigate to the database browsing page.
Tap the table-selection spinner to reveal the list of available tables.
Choose a specific table.
Observe as the app fetches that table's data and reload the recycler view.

Pass/Fail Criteria

Pass: The spinner closes on selection. The recycler view immediately displays one card per record in the chosen table, showing the correct information on data cards. Scrolling through the list is smooth and no placeholders or loading errors appear.
Fail: The recycler view remains empty or shows wrong/mismatched records. Any "No data" message when records exist or any error dialog appears. UI freezes or crashes during data load.

Figure 9 Test Case 5: Load and Display Table Data via Spinner

Test Case 7: DAO Insert Method Coverage

In this scenario we exercise the DAO's insert() method directly in a unit test against an in-memory SQLite database.

Procedure

Set up an in-memory SQLite instance in test code.
Create two item objects: one valid (name = "Test data", quantity = 5, category = "Test") and one invalid (name = "", quantity = –1).
Call dao.insert(validItem) and immediately query via dao.getById() or raw SQL to fetch that row.
Wrap dao.insert(invalidItem) in a try/catch to trigger the constraint violation branch.

Pass/Fail Criteria

Pass: After calling dao.insert(validItem), querying the database returns a row matching all fields of validItem. Inserting invalidItem throws the expected exception (SQLiteConstraintException) or returns the defined error result, and the database state remains unchanged.
Fail: The valid insert does not persist correctly (data mismatch or missing row), or the invalid insert does not raise the expected exception (no error thrown or wrong exception), or the database state is altered by the invalid insert.

Figure 10 Test Case 7: DAO Insert Method Coverage

iii. Testing

Ten test cases were performed. The most important ones are listed in the table below. Most passed. One failed due to missing validation.

For example, Test 1: Checked offline input. Test 2: Checked login session storage. Test 4: Checked profile image upload. Test 7: Failed because some forms accept wrong input.

iv. Test Results

Table 2 Summary of Testing Results

Test Case	Description	Result
TC1	Offline data entry	Passed
TC2	Login session	Passed
TC3	Report generation	Passed
TC4	Profile picture upload	Passed
TC5	User feedback form	Passed
TC6	App behavior after logout	Passed
TC7	Input validation on Edit Form	Failed
TC8	Password reset function	Passed
TC9	Table display on Home	Passed
TC10	Install and upgrade test	Passed

Only Test 7 failed because the form did not check input properly. All other functions worked without major problems. Screenshots and logs were taken during testing.

Based on the feedback collected from several users, the Farmware application still has some areas that can be improved. Many users asked for clearer instructions or a simple guide when using the app for the first time. Some said the offline mode works, but it's hard to know when it is active. There were also requests to make the text size bigger. A few users wanted features like report export and tagging to help manage their data better. One suggestion was to add auto saving, in case users forget to press save. One user also mentioned that the interface looks too plain and could use more color. These responses are shown in figure 11.

Figure 11 Responses of farmware

5.0 CONCLUSION

The development and testing of the Farmware system followed an agile method. This helped manage the project better even with limited resources. It gave a clearer path and allowed the team to finish parts of the system on time. It also made the documentation more structured, which helped in tracking progress and finding issues early. During the process, several problems were found. Some turned into useful features, while others became lessons for improvement.

The final product still has some issues. At the beginning, the system was over-planned, and not all goals could be completed within the timeline. One of the biggest problems is the database structure. The current version does not have foreign keys linking user data to their inventory records. This makes it hard to separate data by user and increases the complexity of some queries. This issue was not planned in the early design.

Another problem is the lack of experience with certain tools, especially the Room library. Some features like hint and submit were built with manual logic, and early attempts to make shared functions failed because of differences in how data is handled across the app.

Some parts of the design were also not based on real user behavior. For example, the input system was built to handle one form at a time, but some cases need multiple steps like choosing, editing, and updating a list of items. This was not included in the original plan, so some features couldn't be built properly.

There are also some planned functions that were not finished. Online mode and data encryption are two of them. The app uses global variables and shared files for now, but this will not be enough if more advanced features are added in the future.

Even with these limits, the app reached some useful goals. It has a simple and clean interface. It supports basic form functions like add, edit, and delete. The hint system gives users feedback for valid input. Performance is good, and the app is suitable for people who are new to digital systems or small farming businesses.

If development continues, the technical issues should be solved first. Then new features like encryption, cloud sync, and even smart tagging or AI support can be added. One possible change is to redesign the database so that each user's data is clearly separated, either by file or by adding a user ID as a link in every table. This is shown in Figure 12.

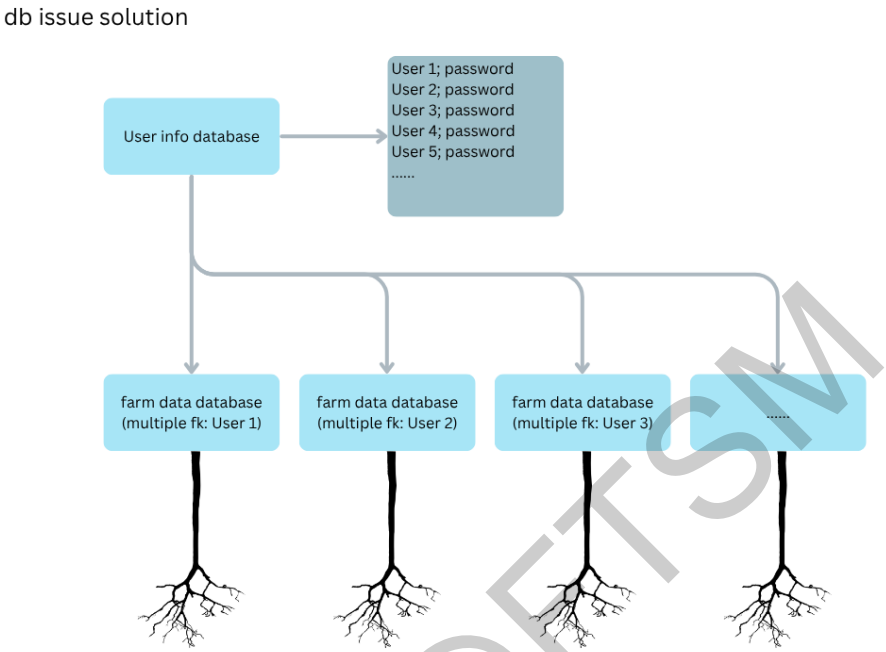


Figure 12 Solution of the database defect

Query logic should also be improved by building shared functions that can be reused. This needs a better understanding of Room and related tools. Security is another key area to fix. User data should be encrypted properly, and access to data must be handled more carefully inside the system.

If these improvements are made, Farmware can become more helpful to small producers. It can support better farm planning, reduce waste, and in the long term, give ideas to other projects looking to solve real world problems through simple and smart tools.

6.0 APPRECIATION

The developer would like to thank the Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia, for providing the learning environment and support throughout the development of this project.

7.0 REFERENCES

6 Farm Information Management Systems Changing The Future of Farming. (2022). https://www.benchmarklabs.com/blog/6-farm-information-management-systems-changing-future-of-farming/

Abbasi, R., Martinez, P., & Ahmad, R. (2022). The digitization of agricultural industry–a systematic literature review on agriculture 4.0. *Smart Agricultural Technology*, *2*, 100042.

Agarwal, S. (2014). Economic order quantity model: a review. *VSRD International Journal of Mechanical, Civil, Automobile and Production Engineering*, *4*(12), 233-236.

Akhter, R., & Sofi, S. A. (2022). Precision agriculture using IoT data analytics and machine learning. *Journal of King Saud University-Computer and Information Sciences*, *34*(8), 5602-5618.

Amini, M., & Rahmani, A. (2023). Agricultural databases evaluation with machine learning procedure. *Australian Journal of Engineering and Applied Science*, *8*(2023), 39-50.

Dewi, L., Wijaya, I., & Seputra, K. (2021). Web-based Buleleng regency agriculture product information system development. Journal of Physics: Conference Series,

Ebbers, T., Kool, R. B., Smeele, L. E., Dirven, R., den Besten, C. A., Karssemakers, L. H., Verhoeven, T., Herruer, J. M., van den Broek, G. B., & Takes, R. P. (2022). The impact of structured and standardized documentation on documentation quality; a multicenter, retrospective study. *Journal of Medical Systems*, *46*(7), 46.

Hameed, M. A., & Arachchilage, N. A. G. (2021). The role of self-efficacy on the adoption of information systems security innovations: a meta-analysis assessment. *Personal and Ubiquitous Computing*, *25*(5), 911-925.

Hasan, Y., Shamsuddin, A., & Aziati, N. (2013). The impact of management information systems adoption in managerial decision making: A review. *The International Scientific Journal of Management Information Systems*, *8*(4), 010-017.

Madamidola, O. A., Daramola, O., Akintola, K., & Adeboje, O. (2024). A Review of Existing Inventory Management Systems. *International Journal of Research in Engineering and Science (IJRES)*, *12*(9), 40-50.

Msangi, J. P., & Msangi, J. P. (2014). Population, Agriculture, Poverty and Food Security: An Overview. *Food Security Among Small-Scale Agricultural Producers in Southern Africa*, 1-19.

Oladele, T. O., Ogundokun, R. O., Adegun, A., Adeniyi, E., & Ajanaku, A. T. (2021). Development of an inventory management system using association rule. *Indonesian Journal of Electrical Engineering and Computer Science*, *21*(3), 1868-1876.

Profile. (2024). https://www.kpkm.gov.my/en/corporate/profile

Tata Sutabri, T. S. (2023). Design of A Web-Based Social Network Information System. *International Journal of Artificial Intelligence Research*, *6*(1), 310-316.

Zheng Bitao (A191802)

Dr. Faizan Qamar
Senior Lecturer
Center for Cyber Security (CYBER),
Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM), 43600 UKM Bangi, Selangor, Malaysia.
Email: faizanqamar@ukm.edu.my ,